Handling Weighted, Asymmetric, Self-Looped, and Disconnected Networks in ORA

Wei Wei, Jürgen Pfeffer, Jeffrey Reminga, and Kathleen M. Carley

August, 2011
CMU-ISR-11-113

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Center for the Computational Analysis of Social and Organizational Systems
CASOS technical report.

This work is part of the Dynamics Networks project at the center for Computational Analysis of Social and Organizational Systems (CASOS) of the School of Computer Science (SCS) at Carnegie Mellon University (CMU). This work is supported in part by the Office of Naval Research (ONR), United States Navy, N00014-08-11223. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **AUG 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE
Handling Weighted, Asymmetric, Self-Looped, and Disconnected Networks in ORA

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carnegie Mellon University,School of Computer Science,Institute for Software Research,Pittsburgh,PA,15213

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

When Linton C. Freeman made his conceptual clarifications about centrality measures in social network analysis in 1979 he exclusively focused on unweighted, symmetric, and connected networks without the possibility of self-loops. Even though a lot of articles have been published in the last years discussing network measures for weighted, asymmetric or unconnected networks, the vast majority of researchers dealing with social network data simplify their networks based on Freeman?s 1979 definitions before they calculate centrality measures. When dealing with weighted and/or asymmetric networks which can have self links and consist of multiple components, researchers are confronted with a lack of standardization. Different tools for social network analysis treat specific cases differently. In this article we describe and discuss the ways the software ORA (developed by CASOS at Carnegie Mellon University) handles the most important network measures in case of weighted, asymmetric, self-looped, and disconnected networks. In the center of our attention are the following measures, degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, and clustering coefficient.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**34**	

Keywords: weighted networks, asymmetric networks, self-looped networks, disconnected networks, degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, clustering coefficient, ORA/UCINET measure comparison

Abstract

When Linton C. Freeman made his *conceptual clarifications* about centrality measures in social network analysis in 1979 he exclusively focused on unweighted, symmetric, and connected networks without the possibility of self-loops. Even though a lot of articles have been published in the last years discussing network measures for weighted, asymmetric or unconnected networks, the vast majority of researchers dealing with social network data simplify their networks based on Freeman's 1979 definitions before they calculate centrality measures. When dealing with weighted and/or asymmetric networks which can have self links and consist of multiple components, researchers are confronted with a lack of standardization. Different tools for social network analysis treat specific cases differently. In this article we describe and discuss the ways the software ORA (developed by CASOS at Carnegie Mellon University) handles the most important network measures in case of weighted, asymmetric, self-looped, and disconnected networks. In the center of our attention are the following measures, degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, and clustering coefficient.

Table of Contents

1 Introduction

To describe the structure of networks or the positions and importance of nodes, a large number of measures can be used. The most used measures are centrality measures which help researchers to identify important nodes. Different centrality measures (Wassermann & Faust, 1995) focus on different aspects of centrality. Freeman (1979) defined “three distinct intuitive conceptions of centrality”, degree centrality, closeness centrality, and betweenness centrality. In this article, Freeman describes these concepts with a very simple network structures (a star) and he uses just networks which are undirected, unweighted, connected and without self-loops.

Researchers who work with networks based on real world data often have different data. Networks based on surveys data are, for example, normally directed. The interviewed persons report the contacts they have from their perceptions. Unless we also interview these alters, we do not know whether these connections are reciprocated and can therefore be interpreted as undirected links. Another area where researchers work with directed networks are communication networks. Every e-mail, phone call, or tweet has a direction from a sender to one or more receivers. These communication networks imply also that the weight of the links is an important issue. When we construct theses networks, we normally aggregate the communication flow of a specific time period (e.g. one day or one week). The results are weighted networks where the link weights represent the number of e-mails sent or the summed minutes of telephone conversation. If we want to look at communication networks at the group level, e.g., to analyze the relations between companies, departments, or squads, then self-loops arise because persons in a group (a node in our network) communicate with other people in the same group. And of course, if the networks are large enough then unconnected components, or unreachable nodes in connected but directed networks, occur.

So, networks which are directed, weighted, unconnected, and which even contain self-loops are not unusual in social network analysis. Nevertheless, more than 30 years after Freeman’s conceptual clarifications article, most of the articles nowadays discussing measures in social network analysis literature close their initial definition section with the following sentence: “For simplicity we focus in our work on unweighted, undirected, and connected networks.” In this article we do the opposite, we focus on weighted, asymmetric, self-looped, and disconnected networks. The following sections of this article discuss these characteristics for degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, and the clustering coefficient.

All these characteristics and options of how to handle these characteristics are implemented in the software ORA (Carley et al., 2010). ORA is a dynamic meta-network assessment and analysis tool developed by CASOS at Carnegie Mellon University. It contains hundreds of social network and dynamic network metrics and methods and has been proved to be a powerful analyzing tool in the network science area. Therefore, we use ORA to show the impact of including or excluding the interested characteristics into network measure calculations. In addition we compare the results with the results calculated by UCINET (Borgatti et al., 2002), which is another powerful and widely-used analyzing tool in area of social network analysis. All the experiments are conduct based on ORA 2.3.5 and UCINET 6.346. In the case studies section you can find the data for the networks we used for the experiments. At the end of the next section you will also find ways to manipulate the discussed characteristics on your networks using ORA.

2 Definitions

In this article we presume readers have a basic knowledge of social network analysis and therefore do not make detailed introductions into the field. If the reader is interested in basics and first steps in social network analysis, we refer to the book by Wasserman and Faust (1995) and by Scott (2000). For an introduction into dynamic meta-networks we refer to Carley (2002).

Social networks can be described as graphs consisting of a set of nodes N and a set of edges E connecting the nodes. We use small letters when we discuss single nodes (e.g., u, v) or edges (e) and the large letters N and E to name the whole sets. The number of nodes in a given networks is denoted with $|N|$ and the number of edges with $|E|$. Network data are represented in matrices. The matrix entry w_{uv} describes the relation from node u to node v. We use the words *edge*, *relation*, and *link* interchangeably. The network characteristics which are discussed in this article describe attributes of the set of edges. In the following paragraphs we define these different characteristics.

2.1 Binary Networks

A binary network is constructed by binary values (either 1 or 0) in its network matrix and contains only the information whether a link between two entities in the network exists or not. In the network matrix, 0 in the cell w_{uv} indicates that there is no links from node u to node v while 1 indicates that there is a link. Because the weights of all links are 1 and therefore equal these networks are also called *unweighted* networks.

2.2 Weighted Networks

If the weights of the links are different we use the term weighted network. In a weighted network every link is represented by a real number w_{uv} (continuously from $-\infty$ to $+\infty$, but we ignore negative line weights in this article) in its network matrix and contains not only the information about whether there is a link between entities, but also numerical information about the links (e.g., how far two entities are distant geographically or how often agents interact with each other). In the network matrix, 0 indicates there is no link between two entities while any value other than 0 indicate there is a link between the entities.

2.3 Self-Looped Networks

Self-looks (also called self-links or loops) are links from a node to itself. A self-looped network has therefore non-zero diagonal elements in the network matrix. Depending on the weight representation of the network (either binary network or weighted network), these diagonal elements can take the values 1 (in a binary network) or it can take any real number (in a weighted network).

2.4 Symmetric/Asymmetric Networks

In a symmetric network for every edge e_{uv} there is also an edge e_{vu}. All links are therefore reciprocal. In asymmetric networks this is not the case. In its matrix representation, a symmetric network has symmetric values about its diagonal. Asymmetric networks are also called directed networks, while undirected networks are synonymous with symmetric networks. In weighted networks the link values of all paired symmetric matrix elements have to have the same value in order to be symmetric.

2.5 Disconnected Networks

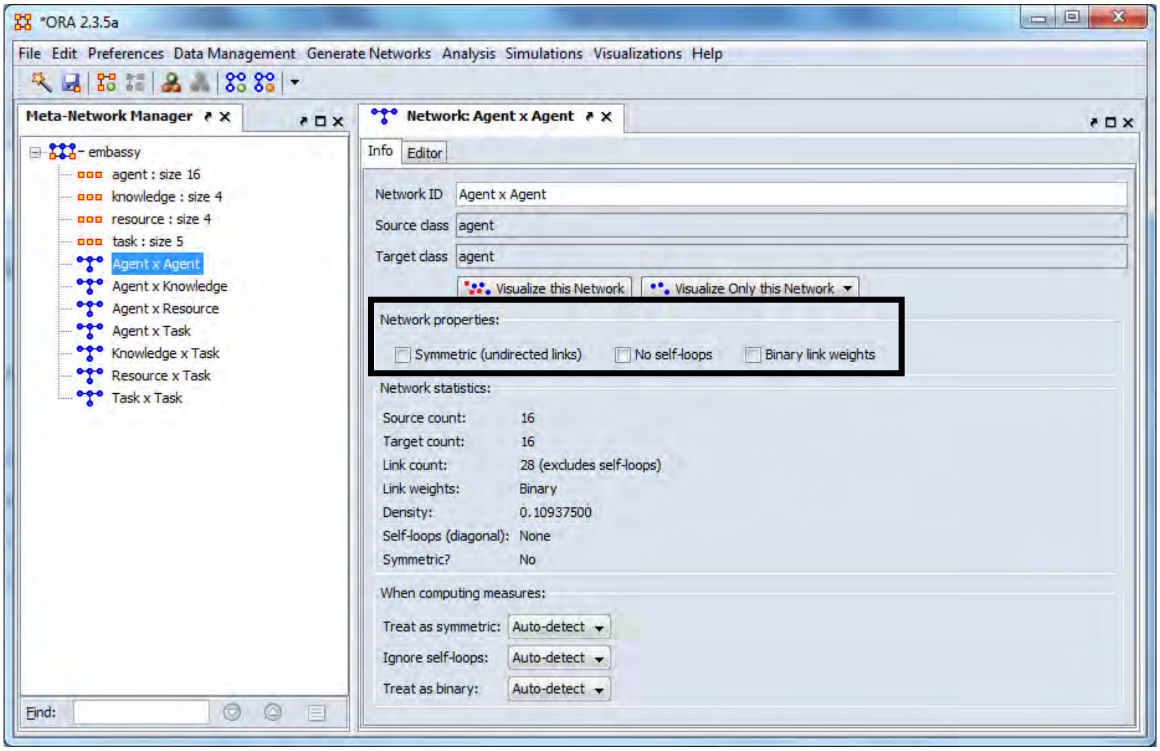
A path in a network is a subset of nodes and edges which connects two nodes without repeating a node or an edge. All nodes which can be reached from a specific node using paths are called reachable. If subsets of nodes are arranged in a way that all nodes of group *A* are unreachable from all nodes from group *B* and vice versa, the network is disconnected. Therefore, there is no link connecting any pair of nodes between the subsets of nodes. These subsets of nodes are named components. Every component can be interpreted as a single network, but researchers are often interested in treating disconnected networks as a single network. In addition, it is important to know that even if a network is connected, it is possible that there are unreachable nodes for a specific node in case of directed networks.

2.6 Network Characteristics in ORA

The network characteristics described in the previous sub-sections change the way a network matrix is composed which has, e.g., implications to different network measures or statistics or the interpretation of the results of network measures (see next section). Therefore, in ORA you have the options to determine these characteristics for your network data. Normally, the person who creates a network knows whether a specific network is directed or undirected etc. Fig. 1 shows a screen shot of these settings in ORA. The options which are selected here are stored as meta-information of every network in the dynetml file. So, if you share your networks with other researchers the selected network characteristics are part of your data.

Changing these options for existing networks can have huge impact to your data. Changing a network from “weighted” to “unweighted” will set all link weights w_{uv} to 1.0. These options also have implications for the editor. If, e.g., the option is set to “directed network” then changing the value w_{uv} in the matrix will automatically change the value w_{vu}.

Independently from these network settings ORA offers additional options for treating these characteristics when network measures are calculated. So, your data can have link weights but it is your decision to ignore these link weights when calculating network measures. You will learn more about these options in the following section.


*ORA 2.3.5a
File Edit Preferences Data Management Generate Networks Analysis Simulations Visualizations Help
Meta-Network Manager
Network: Agent x Agent
Info
Editor
Visualize this Network
Visualize Only this Network
Find:

- embassy
  - agent : size 16
  - knowledge : size 4
  - resource : size 4
  - task : size 5
  - Agent x Agent
  - Agent x Knowledge
  - Agent x Resource
  - Agent x Task
  - Knowledge x Task
  - Resource x Task
  - Task x Task

Network ID: Agent x Agent

Source class: agent

Target class: agent

Network properties:

□ Symmetric (undirected links) □ No self-loops □ Binary link weights

Network statistics:

Source count:	16
Target count:	16
Link count:	28 (excludes self-loops)
Link weights:	Binary
Density:	0.10937500
Self-loops (diagonal):	None
Symmetric?	No

When computing measures:

Treat as symmetric: Auto-detect

Ignore self-loops: Auto-detect

Treat as binary: Auto-detect

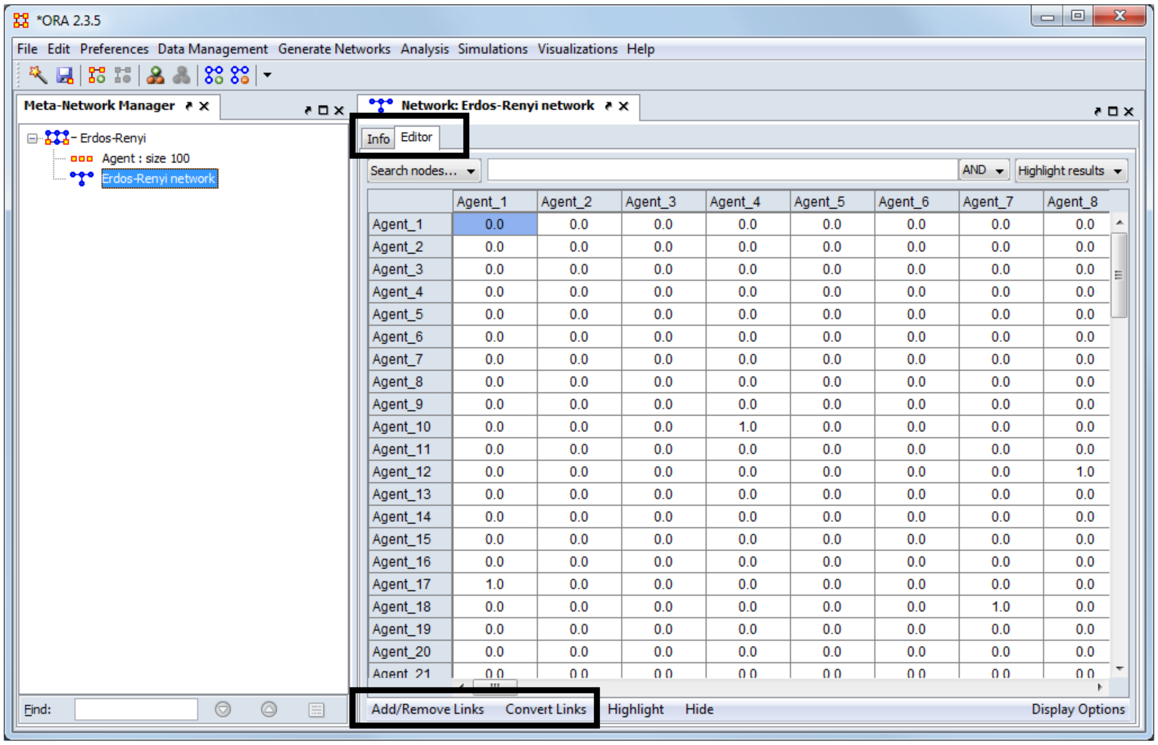
Figure 1: Set network characteristics in ORA

2.7 Manually Manipulate Networks in ORA

Beside the global characteristics which were introduced in the previous sub-section it is also possible to manually manipulate your networks. Fig. 2 shows the additional menu in the editor window of a network. The methods you will find there are very self-explanatory. Here is a quick overview:

Add/Remove Links. With these methods you can *remove* specific links, e.g., links with a line value lower or higher than a given value or self-loops. It is also possible to *set the self-loops* (diagonals) of a network to a designated value. In this menu you can also find ways to *symmetrize* your networks using different methods (maximum, minimum, sum, average).

Convert Links: This menu item includes different ways to manipulate the line weights of your network. You are able to *binarize* all links or just links within a specific range (*collapse*). *Negate* changes the algebraic sign of the links in the network while absolute value turns all line weights to positive numbers. Row-normalize is a method to weight the importance of a single link with the number of links of a node.



	Agent_1	Agent_2	Agent_3	Agent_4	Agent_5	Agent_6	Agent_7	Agent_8
Agent_1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_10	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
Agent_11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Agent_13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_17	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_18	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Agent_19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Agent_21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

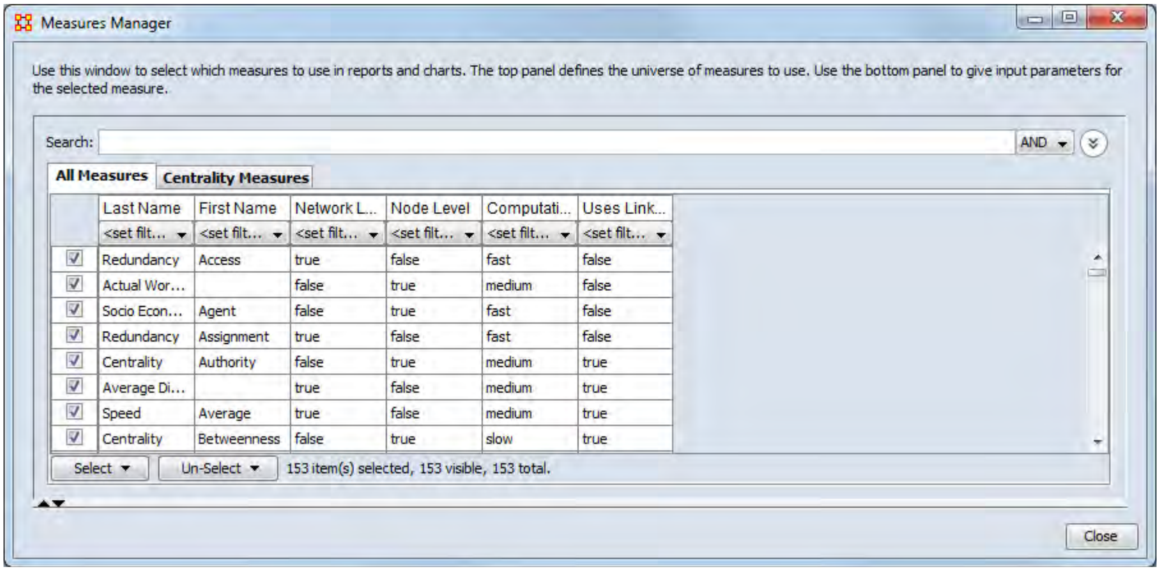
Figure 2: Manually manipulate networks in ORA

3 Network Measures in ORA

Before we start to discuss the impact of the introduced characteristics to different measures, we give a short introduction into the topic of measures in ORA and the options to tell ORA how measures should treat your network data. ORA is designed to handle multi-mode meta-networks. In addition to agents, events, knowledge, locations, resources, and tasks (Carley, 2002) can be analyzed at the same time. This results in a broad variety of network measures. Currently, 152 measures are included in ORA, the standard social network analysis measures as well as measures to analyze different node classes of meta-networks.

3.1 Reports to Generate Measures

There are several different ways to get network measures in ORA. The *normal* way is by using reports. Reports are collections of measures based on different research questions. The *Standard Network Analysis* report includes all measures which are used in this article. To get access to all network measures implemented in ORA you can use ORA's *All Measures* report. You can also use the *All Measures* report to just calculate a selection of measures. To do so, one needs to choose the measures that will be needed in the report before the *All Measures* report is selected. To choose the measure, simply 1) click Analysis in the menu 2) select measure manager. In the pop up window (see fig. 3) you can select or unselect the measures. To better assist finding measures, the measures are grouped in *measure families*. Different families have different *last names*. ORA also provides a search filter and drop down selection filters to find measures more easily. For example, if we want to find degree and betweenness centrality, we can first select centrality in the last name field and then select degree centrality and betweenness centrality in the window. When you are finished with selecting the measure, close the window to save the options. When you now select the *All Measures* report just the measures you selected on the measures manager are calculated.



Use this window to select which measures to use in reports and charts. The top panel defines the universe of measures to use. Use the bottom panel to give input parameters for the selected measure.

Search: AND

	Last Name	First Name	Network L...	Node Level	Computati...	Uses Link...
	<set filt...	<set filt...	<set filt...	<set filt...	<set filt...	<set filt...
☑	Redundancy	Access	true	false	fast	false
☑	Actual Wor...		false	true	medium	false
☑	Socio Econ...	Agent	false	true	fast	false
☑	Redundancy	Assignment	true	false	fast	false
☑	Centrality	Authority	false	true	medium	true
☑	Average Di...		true	false	medium	true
☑	Speed	Average	true	false	medium	true
☑	Centrality	Betweenness	false	true	slow	true

153 item(s) selected, 153 visible, 153 total.

Figure 3: ORA's measure manager.

3.2 Measures as Attributes of Nodes

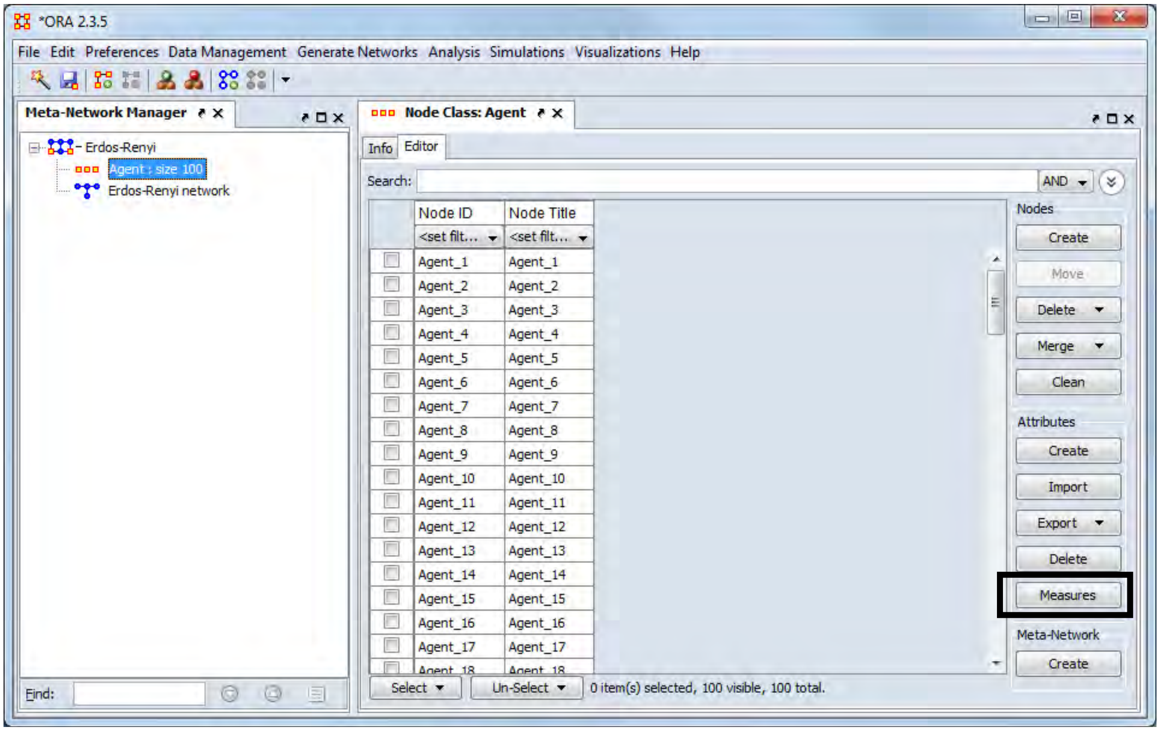
When working with measures in ORA it is important to know that a report never makes any changes to the underlying data. But, sometime you want to calculate a measure for further analysis. In this case you can add the result of a measure as an attribute to the node class for which the measure is calculated (see fig. 4).

It is also possible to create measures in the ORA Visualizer to map, for example, a centrality measure to the size of the nodes. But, we do not discuss this topic in this article. For further details of using ORA we refer to the ORA user manual (Carley et al., 2011).

3.3 Primary Measure Parameters

Several measures, for example those discussed in this article, need different considerations in case of weighted/unweighted, symmetric/asymmetric, connected/disconnected networks or networks allowing/ignoring self-loops. Most of these differences come into play in the context of *normalization* under the different network types which were introduced in section 2. Normalization is the process of making networks of different sizes comparable by dividing the results from the measures by a certain factor which can be very different for different measures. Normalized (also named scaled) values are within the range between 0 and 1. Degree centrality, for example, computed on a network with N nodes is normalized with the maximum number of nodes which could be connected to a single node in a given network, namely, *N-1*. But, this is different if we allow self-loops in a network. Then the number of possible links is

N because every node could also have a link to itself. This normalization factor changes again if we consider directed or weighted networks. We will discuss the different considerations in the context of different measures in the next sections.



	Node ID	Node Title
	<set filt...	<set filt...
□	Agent_1	Agent_1
□	Agent_2	Agent_2
□	Agent_3	Agent_3
□	Agent_4	Agent_4
□	Agent_5	Agent_5
□	Agent_6	Agent_6
□	Agent_7	Agent_7
□	Agent_8	Agent_8
□	Agent_9	Agent_9
□	Agent_10	Agent_10
□	Agent_11	Agent_11
□	Agent_12	Agent_12
□	Agent_13	Agent_13
□	Agent_14	Agent_14
□	Agent_15	Agent_15
□	Agent_16	Agent_16
□	Agent_17	Agent_17
□	Agent_18	Agent_18

0 item(s) selected, 100 visible, 100 total.

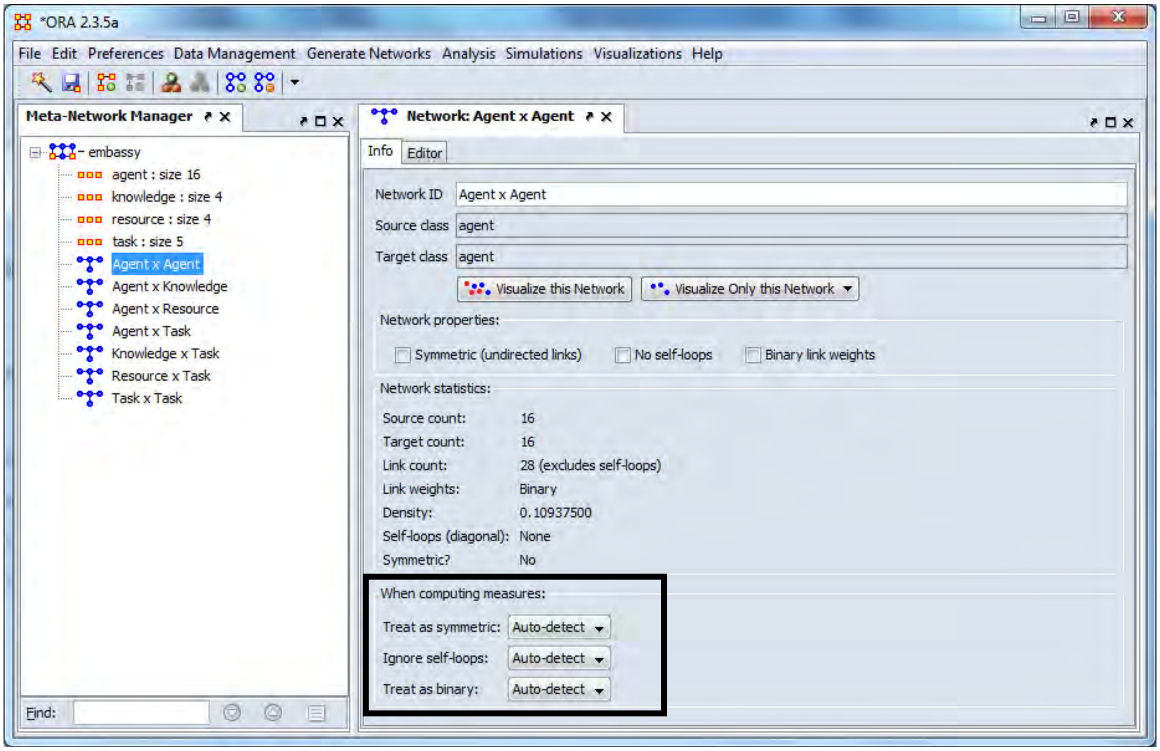
Figure 4: Create measures as node attributes

For a better and more transparent *communication* between the user and ORA, the primary measure parameters are designed as part of every network in ORA. These primary measure parameters, which can be found in the lower part of the info window of a network (see fig. 5), tell ORA what to do with the network matrix before calculating measures. There are three primary measure parameters:

- **Treat as symmetric:** Symmetrizes the network for the calculation, e.g., if the line weight w_{uv} is larger than w_{vu} than $w_{vu} \leftarrow w_{uv}$.
- **Ignore self-loops:** All diagonal elements are set to 0.
- **Treat as binary:** The link weights for all w_{uv} with $w_{uv} \neq 0$ are set to 1.

The default setting of these primary measures is to have ORA auto-detect these settings. For example, if the network is symmetric, then when computing measures the network is considered as symmetric. Similarly, if the network has only binary link weights, then when computing measures the network is considered binary. The user can also explicitly set to True or False whether the network should be treated as symmetric, without self-loops, or binary. To change the settings, select the network and change one or more of the three controls in the info window of a network in the section “Select how to treat the links when computing measures”.

Whenever ORA calculates a measure (independent from which measure calculation you select) a network will be pre-prepared based on the settings of these primary measure parameters. These setting do not change the original data but the way ORA handles the data when calculating measures. To actually convert the data you can change the network parameters (see section 2) or use other procedures to have more detailed options (e.g. symmetrize by minimum value). You can find an introduction into these procedures in section 2.7.


*ORA 2.3.5a
File Edit Preferences Data Management Generate Networks Analysis Simulations Visualizations Help
Meta-Network Manager
Network: Agent x Agent
Info
Editor
Visualize this Network
Visualize Only this Network
Find:

- embassy
  - agent : size 16
  - knowledge : size 4
  - resource : size 4
  - task : size 5
  - Agent x Agent
  - Agent x Knowledge
  - Agent x Resource
  - Agent x Task
  - Knowledge x Task
  - Resource x Task
  - Task x Task

Network ID: Agent x Agent
Source class: agent
Target class: agent

Network properties:

□ Symmetric (undirected links) □ No self-loops □ Binary link weights

Network statistics:

Source count:	16
Target count:	16
Link count:	28 (excludes self-loops)
Link weights:	Binary
Density:	0.10937500
Self-loops (diagonal):	None
Symmetric?	No

When computing measures:

Treat as symmetric:	Auto-detect
Ignore self-loops:	Auto-detect
Treat as binary:	Auto-detect

Figure 5: Change the primary measure parameters

3.4 Scaling Parameter

The results of centrality and other measures in ORA are normally within the range of 0 and 1. This is the result of a scaling procedure to make networks with different sizes comparable. We discuss the scaling of different measures in the following sections. At this place we just want to mention that there is an option in ORA to scale the results within the range of 0 and 100. This percentage scaling is preferred by some scientists. This option can be found in ORA under Preferences>Measures>Scale measures as percentage.

3.5 Impact of Network Characteristics to Measures

In the last sections we introduced different characteristics of network data and the options to determine ORA’s handling of your network data. Table 1 shows an overview of the impact of network characteristics on the network measures which are discussed in this article. The first three of these characteristics are identical with primary measure parameters from the previous sub-section, e.g. when you select the option “Ignore self-loops" this will affect the result of the calculation of degree and eigenvector centrality as well as of the clustering coefficient in case your network contains information on self-looped edges. In the next section we discuss these characteristics with the different measures. In section 9 you can find some case studies where we show the impact of these characteristics to different networks. The case studies also include comparisons between the results of ORA and of UCINET and discusses any calculation differences.

Table 1: Characteristics of networks and their impact on measures

Measure	Allow/Ignore Self-Loops	Symmetric/ Asymmetric	Binary/ Weighted	Connected/ Disconnected
Degree Centrality	Yes	Yes	Yes	No
Betweenness Centrality	No	Yes	Yes	No
Closeness Centrality	No	Yes	Yes	Yes
Eigenvector Centrality	Yes	(No)	Yes	Yes
Clustering Coefficient	Yes	Yes	No	No

If any option is set to True or False, then ORA will ensure by adding/removing links that the property holds in the network. Auto-detect sets the property to True or False based on the existing links; Auto-Detect never adds/removes links when preparing the networks for calculations.

4 Degree Centrality

Technical name: Degree Centrality
Commonsense name: In The Know
Main reference: Freeman (1979)
Maximum theoretical: scaled: 1.0 unscaled: see Table 1
Minimal theoretical: scaled: 0.0 unscaled: 0.0
Description: Degree centrality measures the number of other nodes that one node is connected to. Depending on the network, high degree centrality indicates a highly active agent or an agent known by a lot of other agents, etc.

Table 1 showed us that degree centrality is affected by self-looped, directed, and weighted edges. In the case of a directed network, degree centrality can be separated into three sub measures: In-degree centrality (the number of nodes that point to the entity), out-degree centrality (the number of nodes that the entity points to) and total degree centrality (the number of nodes that both the entity points to and receives from). We therefore handle the characteristic of directed links by separating the considerations on degree centralities into these three groups. The characteristic of weighted links is treated with the following consideration. Instead of counting the number of neighbors for a node v we summarize the link weights w_{vu} and/or w_{uv} to and/or from these neighbors[1]. Finally,

[1] Summing the line weights can be seen as a generalization of counting the lines in the unweighted case. If we define the line weights with 1 in the unweighted case, we are also able to calculate the degree by summing up the line weights. Therefore, no differentiation in the algorithmic implementation is needed for weighted/unweighted networks.

the possibility of a self-loop of a node increases the number of cells in the matrix to look for a value by 1.

4.1 Unscaled Degree Centrality

The unscaled degree centrality C_D counts the absolute number of neighbors in the unweighted case or sums up the line weights connected to every node. Eq. 1 shows the definition of the unscaled degree centrality in symmetric networks. Eq. 2 shows the formula for unscaled in-degree centrality for asymmetric networks, eq. 3 for out-degree centrality, and eq. 4 for the total degree centrality in asymmetric networks.

$$C_D(u) = \begin{cases} \sum_{\mathrm{v}=1,\mathrm{v}\geq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{v,u}} & \text{allow self-loops} \\ \sum_{\mathrm{v}=1,\mathrm{v}>u}^{|\mathrm{N}|} w_{\mathrm{v,u}} & \text{ignore self-loops} \end{cases} \tag{1}$$

$$C_{D\_in}(u) = \begin{cases} \sum_{\mathrm{v}=1}^{|\mathrm{N}|} w_{\mathrm{v,u}} & \text{allow self-loops} \\ \sum_{\mathrm{v}=1,\mathrm{v}\neq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{v,u}} & \text{ignore self-loops} \end{cases} \tag{2}$$

$$C_{D\_out}(u) = \begin{cases} \sum_{\mathrm{v}=1}^{|\mathrm{N}|} w_{\mathrm{u,v}} & \text{allow self-loops} \\ \sum_{\mathrm{v}=1,\mathrm{v}\neq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{u,v}} & \text{ignore self-loops} \end{cases} \tag{3}$$

$$C_{D\_total}(u) = \begin{cases} \left(\sum_{\mathrm{v}=1,\mathrm{v}\neq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{v,u}} + \sum_{\mathrm{v}=1,\mathrm{v}\neq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{u,v}}\right) + w_{\mathrm{u,u}} & \text{allow self-loops} \\ \sum_{\mathrm{v}=1,\mathrm{v}\neq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{v,u}} + \sum_{\mathrm{v}=1,\mathrm{v}\neq \mathrm{u}}^{|\mathrm{N}|} w_{\mathrm{u,v}} & \text{ignore self-loops} \end{cases} \tag{4}$$

The possibility of the self-loop results in the fact that in these networks the sum of the in-degree and the out-degree is different from the total-degree because the self-loop is just counted one time.

4.2 Scaled Degree Centrality

Unscaled degree centrality provides the information about the sum of the line weights for every node. This result is dependent on the number of nodes, e.g., in a network with 100 nodes every node has an unscaled degree centrality within the range of 0 to 100 if the network is unweighted. On the other hand, in a smaller network consisting of just 10 nodes, the unscaled degree centrality of the most important actor is constrained to that number. To make networks with different sizes comparable, we scale (or normalize) the results of the unscaled degree centrality resulting in the scaled degree centrality.
The idea of scaling for the degree centrality is that the values of all nodes are divided by a scaling factor, which is the maximum possible value of these measures (see eq. 5).

$$C'_D(u) = \frac{C_D(u)}{{C_D}^{max}} \tag{5}$$

Looking at eq. 5, $C_D(u)$ could be the unscaled degree of the symmetric network or any unscaled version for asymmetric degree centrality, $C_{D\_in}(u)$,$C_{D\_out}(u)$, or $C_{D\_total}(u)$.

${C_D}^{max}$ is the maximum possible value of the selected degree centrality measure for the given network. This value is different when considering the different network characteristics. Based on table 1 we have to define the scaling factor for networks having self-looped, directed, and weighted edges, or not. For different combinations of these characteristics the scaling factor is different. Once again, in case of asymmetric networks we separate the degree measure into in-degree, out-degree, and total degree. Table 2 shows the scaling factor for the scaled degree centrality for the different combinations of the dependent characteristics. $|N|$ stands for the number of nodes in the network, $w*$ is the maximum value of all link values in the network. A node without any links (isolate) has a degree centrality of 0.

Table 2: Scaling factor for scaled degree centrality, identical to the maximum possible unscaled degree centrality

	Symmetric/ Asymmetric	Allow/Ignore Self-Loops	Binary	Weighted
C_D	symmetric	ignore	$\|N\|-1$	$(\|N\|-1) \cdot w*$
C_D	symmetric	allow	$\|N\|$	$\|N\| \cdot w*$
C_{D_in} or C_{D_out}	asymmetric	ignore	$\|N\|-1$	$(\|N\|-1) \cdot w*$
C_{D_in} or C_{D_out}	asymmetric	allow	$\|N\|$	$\|N\| \cdot w*$
C_{D_total}	asymmetric	ignore	$2\|N\|-2$	$(2\|N\|-2) \cdot w*$
C_{D_total}	asymmetric	allow	$2\|N\|-1$	$(2\|N\|-1) \cdot w*$

The case for binary networks without self-loops is defined by Freeman (1979). Freeman (1979) describes the maximum possible value for degree centrality in binary unweighted network as the center of a star with one node in the middle which is connected to all other nodes. The central node therefore has an in- and out-degree centrality of $|N|-1$ because this is the number of other nodes in the network. Because we count every connection in an undirected network in both directions the scaling factor for the total degree is twice the scaling factor of in- and out-degree. If we allow self-loops the number of possible links increases by one – the self-loop. In case of weighted networks the scaling factor for the binary networks is multiplied with a factor $w*$ which represents the maximum line value in the overall matrix. Doing so, we can guarantee results within the range of 0 and 1 even if the line weights are higher than 1. This is also important when dealing with very small line weights with values smaller than 1. In these cases the unscaled degree centrality results in very small numbers.

For a better understanding of the scaling factors enumerated in table 2 the reader can re-construct these factors with answering the following two questions:

1. How many cells in the network matrix are affected by the calculation (described with the network size $|N|$)?
2. What is the maximum value in the network matrix ($w*$)?

In case of a binary networks question number two results in the value 1 and therefore w* is canceled in table 2 for the binary factors.

4.3 Network Level Degree Centrality

Freeman (1979) also defines the *centralization* of a network. This value gives an impression about the distribution of the centrality values. If the centrality scores are very high for some nodes and very small for the vast majority of the nodes this value is much higher than in cases the centrality scores are almost equally distributed. The network level degree centrality is defined as the sum of differences between the most central node and all other nodes, divided by a possible maximum of this sum of differences (eq. 6). For calculating the network level degree centrality we use the unscaled scores of degree centrality. Using the scaled degree centrality scores would result in the same network level value, but additional scaling would be necessary.

$$C_D = \frac{\sum_{u=1}^{u=|N|} C_D(u^*) - C_D(u)}{max(\sum_{u=1}^{u=|N|} C_D(u^*) - C_D(u))} \qquad (6)$$

In eq. 6 $C_D(u)$ could be the symmetric or one of the three asymmetric unscaled degree centrality measures. $C_D(u^*)$ denotes the maximum value of the specific degree centrality of all nodes in the network. Freeman (1979) showed that the maximum possible value can be achieved in a star like network (similar to the scaling factor for scaled degree centrality). In the binary case when self-loops are ignored, the center of a star has a degree of $|N|-1$ and every other node has a degree of 1 resulting in a maximum value for eq. 6 of $(|N|-1) \cdot (|N|-2)$.Table 3 shows this maximum possible value for all combinations of the affected network characteristics. Again, the weighted case is created by multiplying the binary factor with the maximum line weight w^* in the given network matrix.

Table 3: Maximum possible value for calculating network level degree centrality

	Symmetric/ Asymmetric	Allow/Ignore Self-Loops	Binary	Weighted
C_D	symmetric	ignore	$(\|N\|-1) \cdot (\|N\|-2)$	$(\|N\|-1) \cdot (\|N\|-2) \cdot w^*$
C_D	symmetric	allow	$(\|N\|-1) \cdot (\|N\|-1)$	$(\|N\|-1) \cdot (\|N\|-1) \cdot w^*$
C_{D_in} or C_{D_out}	asymmetric	ignore	$(\|N\|-1)^2$	$(\|N\|-1)^2 \cdot w^*$
C_{D_in} or C_{D_out}	asymmetric	allow	$(\|N\|-1) \cdot \|N\|$	$(\|N\|-1) \cdot \|N\| \cdot w^*$
C_{D_total}	asymmetric	ignore	$(\|N\|-1) \cdot (2\|N\|-4)$	$(\|N\|-1) \cdot (2\|N\|-4) \cdot w^*$
C_{D_total}	asymmetric	allow	$(\|N\|-1) \cdot (2\|N\|-3)$	$(\|N\|-1) \cdot (2\|N\|-3) \cdot w^*$

5 Betweenness Centrality

Technical name:	Betweenness Centrality	
Commonsense name:	Broker, Connector	
Main reference:	Freeman (1977, 1979)	
Maximum theoretical:	scaled: 1.0	unscaled: see Table 4
Minimal theoretical:	scaled: 0.0	unscaled: 0.0
Description:	Betweenness centrality measures the amount an actor is in an intermediate position between other nodes. High between actors connect different groups and have control over the flow of information in a network.	

Anthonisse (1971) saw the *rush* in a graph as the amount an agent in a network has to intermediate between other agents. Freeman (1977, 1979) defined betweenness centrality as one of the “three distinct intuitive conceptions of centrality” (Freeman, 1979: 215). Betweenness centrality is often connected with the notion of control over the flow of information. Betweenness centrality is calculated by a breath-first search algorithm which calculates the shortest paths from every node to all other nodes (Brandes 2001). The nodes which lie on these shortest paths are favored in the betweenness centrality score. Based on table 1 we have to consider the network characteristics of directed and weighted edges for calculating betweenness centrality. So, even though betweenness centrality is more complex than degree centrality, less network characteristics influence the result of betweenness centrality. The weights in a network are treated as distances in ORA when calculating the shortest paths through the network. Therefore, a path $a – b – c$ connected by two edges with line value of 1 is shorter than a path $a – d$ if the line value of this single edge is, e.g., 3.

Before we start to discuss the impact of symmetric/asymmetric networks to betweenness centrality, we want to point the reader to an implication of the handling of line weights in ORA. In social network analytical research projects line weights are used in two different ways. First, as distances to describe, e.g., physical distances, time which information takes from node *a* to node *b*, or dissimilarities of node attributes. Second, as similarities to describe, e.g., the amount of interaction between nodes, emotional nearness, or the similarities of nodes. As mentioned in the previous paragraph, ORA treats line weights as distances. If the line weights of your data represent similarity information, you could either ignore the line weights when calculating betweenness centrality or transform the line weights, e.g., by subtracting from (w^*+1). Note, that subtracting from w^* would result in 0 values which are interpreted as the absence of a line. Another way to transform your data is to take the inverse ($1/w$) of every line weight > 0. However, you should be careful of the implications of applying betweenness centrality calculations to weighted networks.

5.1 Unscaled Betweenness Centrality

Betweenness centrality counts the number of shortest paths through a specific node and weights that path with the number of alternative existing shortest paths. Eq. 7 shows the definition of betweenness centrality. The two summations describe that the shortest paths are calculated from every node to every other node. $g_{u,v}$ is the number of shortest paths between two nodes u and v while $g_{u,v}(k)$ is the number of shortest paths including node k. For symmetric networks Freeman (1979) defined betweenness centrality by just looking at the shortest paths from one half of the matrix. In asymmetric networks the results for the second half are different from the first, therefore, $\frac{g_{u,v}(k)}{g_{u,v}}$ and $\frac{g_{v,u}(k)}{g_{v,u}}$ are not identically and have to be calculated separately.

$$C_B(k) = \begin{cases} \sum_{u=1}^{|V|} \sum_{v=u+1}^{|V|} \frac{g_{u,v}(k)}{g_{u,v}} & \text{symmetric networks} \\ \sum_{u=1}^{|V|} \sum_{v \neq u}^{|V|} \frac{g_{u,v}(k)}{g_{u,v}} & \text{asymmetric networks} \end{cases} \tag{7}$$

5.2 Scaled Betweenness Centrality

The unscaled betweenness centrality scores are generated by counting the number of times a node lies on the shortest paths of other nodes. To scale these values into the range of 0 and 1 which makes the results independent from the network size, we have to divide $C_B(k)$ by the maximum possible score ${C_B}^{max}$ (eq. 8). $C'_B(k)$ denotes the scaled betweenness centrality of node k.

$$C'_B(k) = \frac{C_B(k)}{{C_B}^{max}} \tag{8}$$

The formulas to calculate the maximum possible values for symmetric and asymmetric networks are listed in table 4 and describe the node in the center of a star network (Freeman, 1977, 1979). The maximum score is a function of the number of total nodes of the network |*N*| and can be calculated by looking at all possible combinations of two nodes excluding the center of the stars.

Table 4: Scaling factor for scaled betweenness centrality, identical to the maximum possible unscaled betweenness centrality

	Symmetric	Asymmetric
${C_B}^{max}$	$\frac{\|V\|^2-3\|V\|+2}{2}$	$\|V\|^2 - 3\|V\| + 2$

5.3 Network Level Betweenness Centrality

The network level betweenness centrality of a network is defined the same way as the network level degree centrality measure (see eq. 6). We calculate the network level measure using the unscaled betweenness centrality node level scores. In eq. 9 $C_B(u\,*)$ denote the maximum betweenness centrality score of all the nodes in the network. Table 5 shows the formulas to calculate the divisor of eq. 9.

$$C_B = \frac{\sum_{u=1}^{u=|N|} C_B(u^*) - C_B(u)}{max\left(\sum_{u=1}^{u=|N|} C_B(u^*) - C_B(u)\right)} \tag{9}$$

Table 5: Maximum possible value for calculating network level betweenness centrality

	Symmetric	Asymmetric
$max\left(\sum_{u=1}^{u=\|N\|} C_B(u^*) - C_B(u)\right)$	$\frac{\|V\|^2-3\|V\|+2}{2} \cdot (\|V\|-1)$	$(\|V\|^2 - 3\|V\| + 2) \cdot (\|V\| - 1)$

6 Closeness Centrality

Technical name:	Closeness Centrality	
Commonsense name:	-	
Main reference:	Freeman (1979)	
Maximum theoretical:	scaled: 1.0	unscaled: see Table 7
Minimal theoretical:	scaled: 0.0	unscaled: 0.0
Description:	Closeness centrality measures the nearness (as opposite from the distance) from an agent to all other agents. Agents having a high closeness score have short distances to all other nodes. This is important for the availability of knowledge and resources.	

Sabidussi (1966) described the sum of the shortest path distances from one node to every other node as the node’s farness. Freeman (1979) used this idea to define closeness centrality of a node as the inverse of Sabidussi’s farness. Nodes having a high closeness centrality are nearby all other nodes and have advantages in accessing resources in a network or having a good overview of the agents in a network. Table 1 shows that closeness centrality is affected by directed and weighted edges as well as by the question if the network is connected or not. In case of directed networks the closeness centrality defined by Freeman (1979) can be interpreted as out-closeness centrality because the calculation of the shortest paths follows the links just in the outgoing direction. Consequently we defined in-closeness by following the links in the opposite direction. In-closeness centrality is the closeness is the reachability of an actor from the perspective of all other nodes. The characteristic of line weights is handled similar than for betweenness centrality (as distances). The most interesting related characteristic in the context of closeness centrality is the question whether the network is connected, or not. Freeman states in the context of closeness centrality, that “it is, of course, only meaningful for a connected graph” (Freeman, 1979: 225) because the distance of unreachable nodes is infinitely. We handle this fact by introducing penalty scores for unreachable nodes.

6.1 Unscaled Closeness Centrality

As mentioned above the closeness centrality of a node *k* is the inverse of the sum of the shortest path distances $d_{k,u}$ to all other nodes (eq. 10). In case of directed networks, in-closeness centrality sums the shortest distances from all other nodes $d_{u,k}$ to node *k.* In-closeness centrality is equivalent to out-closeness centrality from the transposed network matrix.

$$C_C(k) = C_{C\_out}(k) = \frac{1}{\sum_{u=1}^{|V|} d_{k,u}} \quad (10)$$

$$C_{C\_in}(k) = \frac{1}{\sum_{u=1}^{|V|} d_{u,k}} \quad (11)$$

The shortest path distance is just defined for nodes which are actually somehow connected through paths because if a node u is unreachable for node k the distance is infinitely. To be able to calculate closeness centrality also in unconnected networks, we introduce a penalty value for unreachable nodes. The idea is to add a value which is higher than the maximum possible distance in the network. The maximum possible distance in a connected and binary network is $|N|$-*1* in case all nodes are arranged on a line where every node is just connected with its neighbors. Therefore, we use $|N|$ as the penalty value. In weighted networks we have to multiply the number of nodes with the maximum line value in the matrix to ensure that no single shortest path could exceed this value. These penalty values are listed in table 6.

Table 6: Penalty values for unreachable nodes for closeness centrality

	Binary	Weighted
Penalty	$\|N\|$	$\|N\| \cdot w^*$

6.2 Scaled Closeness centrality

Like other centrality measures, closeness centrality is scaled by dividing with the maximum possible value of the centrality (eq. 12).

$$C'_C(k) = \frac{C_C(k)}{C_C{}^{max}} \quad (12)$$

In eq. 12 $C_C{}^{max}$ denotes the maximum possible value of the closeness centrality calculation. This maximum possible value could be found, once again, in the center of a start like network. Table 7 shows the scaling factor for binary and weighted networks. w^- is a representation of the minimum line weight in the network (the smallest non 0 value in the network matrix). So, in weighted networks the shortest possible paths could be constructed in case one node is connected directly to all other nodes with the minimum possible path distance.

Table 7: Scaling factor for scaled closeness centrality identical to the maximum possible unscaled closeness centrality

	Binary	Weighted
${C_C}^{max}$	$\frac{1}{\lvert V\rvert-1}$	$\frac{1}{(\lvert V\rvert-1)\cdot w^{-}}$

6.3 Network Level Closeness centrality

Freeman (1979) defines the networks level closeness centrality using the scaled values of closeness centrality. Therefore, we do it the same way, even though, the formulas could be transformed easily and the results are identical. The network level closeness centrality follows the same formula like in the degree and betweenness case (eq. 13).

$$C_C = \frac{\sum_{u=1}^{u=|N|} C'_C(u^*) - C'_C(u)}{max\left(\sum_{u=1}^{u=|N|} C'_C(u^*) - C'_C(u)\right)} \qquad (13)$$

The maximum possible value for the network level measure is defined in eq. 14. Because we use the scaled values for calculating the network level closeness centrality it is not necessary to have separate scaling factors for binary and weighted networks.

$$max\left(\sum_{u=1}^{u=|N|} C'_C(u^*) - C'_C(u)\right) = \frac{|V|^2-3|V|+2}{2|V|-3} \qquad (14)$$

7 Eigenvector Centrality

Technical name: Eigenvector Centrality
Commonsense name: -
Main reference: Bonacich (1972)
Maximum theoretical: scaled: 1.0 unscaled: $\sqrt{0.5}$
Minimal theoretical: scaled: 0.0 unscaled: 0.0
Description: Eigenvector centrality is based on eigenvector calculation in linear algebra. Agents have a high eigenvector score if they are important and connected to other important agents.

Beside the three basic centrality measures which were introduced by Freeman (1979) an additional fourth one is widely used, eigenvector centrality. Bonacich (1972) offers a centrality measure based on the algebraic method of eigenvector calculation. Eigenvector centrality is often connected with the notion of power or the idea that a node is important if it is connected to other important nodes. While degree centrality rewards all links equally, eigenvector centrality makes differences by including also the links of the neighbors and of the neighbors of the neighbors etc.

When looking at table 1, we can see that Eigenvector centrality is affected by almost all network characteristics. The “No” in brackets for symmetric/asymmetric networks results from the fact that eigenvector centrality should not be calculated with asymmetric networks because of the possibility of complex eigenvalues. Therefore, ORA automatically symmetrizes every network for the calculation of eigenvector centrality to guarantee real number results independently from the options described in section 2 and 3. Self-loops and line weights are automatically handled by the algorithm to calculate different results for networks with these characteristics. The remaining characteristic we have to deal with is the case of unconnected networks.

The case of unconnected networks is very tricky for eigenvector centrality because the results are all but intuitive in unconnected networks. Often the node scores in one or more components are all zero without the guarantee that nodes in the largest component actually get non-zero scores. Another oddity when looking at the results of eigenvector centrality without being a mathematician is that the highest score is given to a dyad (2-node component), whereas one is usually interested in finding nodes embedded in larger components – which most likely have scores lower than those of the dyadic component.

Because of these considerations, ORA offers two different eigenvector centrality calculations in case of unconnected networks. First, the standard eigenvector centrality with all the implications discussed in the previous paragraph. Second, eigenvector centrality per component which runs eigenvector centrality on each component independently; this means, extract each component one at a time and make it its own network, call eigenvector centrality and place the scores into a single result vector. To take the different component sizes into account the result value for every node is normalized (in addition to the normalization described in sub-section 7.2) with the component size by $|N_i|/|N|$ where $|N_i|$ is the size of the component including node i. In both cases the networks are symmetrized with the union/maximum method for the calculation. Both eigenvector measures are part of the key entity and the SNA report.

7.1 Unscaled Eigenvector Centrality

The Eigenvector centrality of a node u, $C_E\ (u)$ is defined as the linear combination of the eigenvector centrality of its neighbors:

$$C_E\ (u) = \frac{1}{\lambda}\sum_{v=1}^{|V|} w_{u,v} C_E(U) \qquad (15)$$

where λ is a constant. We can rewrite the equation as:

$$\lambda C_E = W \cdot C_E \qquad (16)$$

In eq. 16, C_E is an eigenvector and W is the network matrix. For calculating eigenvector centrality λ is the largest eigenvalue of the adjacency matrix W and C_E is the corresponding eigenvector. Note that W is always symmetrized before computing the measure which guarantees real (rather than complex) valued eigenvalues.

7.2 Scaled Eigenvector Centrality

Scaling eigenvector centrality follows the same logic as we discussed in the previous sections for the other centrality measures. But, instead of the star like network with the highest possible score, the maximum possible value of eigenvector centrality occurs when the network consists of a single dyad.

$$C'_E(k) = \frac{C_E(k)}{{C_E}^{max}} \qquad \text{with } {C_E}^{max} = \sqrt{0.5} \tag{17}$$

Independently from the network size, the maximum value is always $\sqrt{0.5}$. Consequently, we use this value to scale the unscaled values of eigenvector centrality.

7.3 Network Level Eigenvector Centrality

For the network level of eigenvector centrality, once again, we have to take into considerations the maximum possible differences between the node with the highest score and all other nodes.

$$C_E = \frac{\sum_{u=1}^{u=|N|} C_E(u^*) - C_E(u)}{max\left(\sum_{u=1}^{u=|N|} C_E(u^*) - C_E(u)\right)} \tag{18}$$

This maximum difference can be achieved in a network with a single dyad and no other links which leads to eq. 19:

$$max\left(\sum_{u=1}^{u=|N|} C_E(u^*) - C_E(u)\right) = \sqrt{0.5} \cdot (|N| - 2) \tag{19}$$

8 Clustering Coefficient

Technical name: Clustering Coefficient
Commonsense name: -
Main reference: Watts and Strogatz (1998)
Maximum theoretical: 1.0
Minimal theoretical: 0.0
Description: The Clustering coefficient measures the local density of every agent. Agents with a high clustering coefficient are connected to neighbors which are more likely connected to each other.

In the previous sections we discussed how to handle the four most important centrality measures in case of weighted, asymmetric, self-looped, and disconnected networks. The last measure we want to discuss in this report is the clustering coefficient. The clustering coefficient is not a centrality measure. It describes the density of the ego-network for every node. Watts and Strogatz (1998) used this measure to discuss a very important characteristic of real world social networks consisting of humans, the tendency that there is a higher probability that the nodes which have the same neighbors are connected with each other. The local density of social networks is also connected to the idea of weak and strong ties (Granovetter, 1973). While strong ties are more likely strongly embedded in the social network of a person, weak ties often reach into different areas of the network. The fraction of these strong and weak ties of a node influences the clustering coefficient of a node.

Because the clustering coefficient can be described as a local density measure, table 1 tells us that it is affected by two network characteristics, self-loops, and symmetric/asymmetric links.

8.1 Node Level Clustering Coefficient

The density of a network (Wasserman & Fast, 1995) is defined as the number of actual links in a network divided by the number of possible links. As the clustering coefficient for a node is the density of the ego-network of this node (without the node itself). Watts and Strogatz (1998) defined the clustering coefficient *CC(v)* of a node *p* in simple networks as follows. For a vertex v with k_v neighbors, these neighbors can have at most $k_v*(k_v - 1)/2$ edges. The clustering coefficient for the node v is the number of actual links between the k_v neighbors divided by the maximum possible number. To expand this concept to directed networks which can have self-loops we generalize the equation from Watts and Strogatz to

$$CC\ (u) = \frac{|E_{v,w}|}{|E_{v,w}|^*} \quad with\ e_{u,v}, e_{u,w} \in E. \tag{20}$$

$|E_{v,w}|$ is the number of actual links between the neighbors of u. $|E_{v,w}|^*$ is the number of maximal possible links between these neighbors which is a function of the number of neighbors $|N_u|$ of the node u and the characteristics of the network. Table 8 shows the calculation of $|E_{v,w}|^*$ for the combinations of symmetric/asymmetric and allow/ignore self-loops. In asymmetric networks the whole sub-matrix is part of the calculation while in the symmetric case just one half is considered. The right column of table 8 is the left column increased by the diagonal elements of the matrix.

Table 8: The maximum possible links to calculate the clustering coefficient

	Ignore Self-Loops	Allow Self-Loops
Symmetric	$\frac{\vert N_u\vert \cdot (\vert N_u\vert - 1)}{2}$	$\frac{\vert N_u\vert \cdot (\vert N_u\vert - 1)}{2} + \vert N_u\vert$
Asymmetric	$\vert N_u\vert^2 - \vert N_u\vert$	$\vert N_u\vert^2$

Table 8 describes also which cells of the sub-matrix are included in the calculation of $|E_{v,w}|$. In case of networks with self-loops the results of the clustering coefficient calculation can be all but obvious at first sight (see case studies in section 9).

8.2 Graph Level Clustering Coefficient:

The network level measure for the clustering coefficient is easily defined as the average clustering coefficient of all node level scores in the network:

$$CC = \frac{1}{|N|} \sum_{u=1}^{u=|N|} CC(u) \tag{21}$$

9 Case Studies

9.1 Example Network

In the following pages we show the results of measure calculations discussed in this article. We therefore construct a small network consisting of 6 nodes which covers the different network characteristics. Figure A1 shows a visualization of this case study network. In table A1 the matrix representation of this network can be found. The network is *weighted*, *directed*, contains *self-loops*, and consists of two *unconnected* components. For every measure, we first calculate the centrality measure considering all these network characteristics by setting the primary measure parameters to “Auto-detect” (see section 3.3). Second, we tell ORA to ignore the characteristics one after the other by setting the specific parameter to “True”; we also calculate the measure for the case of ignoring all characteristics at once (the simple network). Columns which are drawn with a gray background are those which affect the measure calculation (see table 1). The characteristic “multiple components” is not a primary measure parameter, but it influences the results of two measures. Third, we calculate the measures with UCINET to compare the results with the ORA results and discuss possible difference. All calculations are accomplished with ORA 2.3.5 and UCINET 6.346. The results in UCINET are in the range [0-100]; the ORA scale range is [0-1]. To change the scale range of ORA see section 3.4.

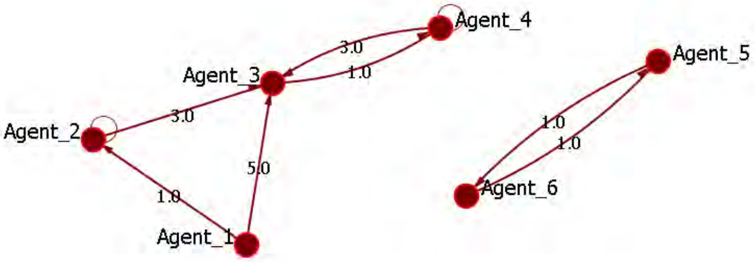


Figure A1: Network for the case studies

Table A1: Matrix form of the network for the case studies

	Agent_1	Agent_2	Agent_3	Agent_4	Agent_5	Agent_6
Agent_1	0.0	1.0	5.0	0.0	0.0	0.0
Agent_2	0.0	1.0	3.0	0.0	0.0	0.0
Agent_3	0.0	0.0	0.0	1.0	0.0	0.0
Agent_4	0.0	0.0	3.0	1.0	0.0	0.0
Agent_5	0.0	0.0	0.0	0.0	0.0	1.0
Agent_6	0.0	0.0	0.0	0.0	1.0	0.0

9.2 Degree Centrality

The characteristics which affect degree centrality are *allow/ignore self-loops*, *symmetric/asymmetric*, and *binary/weighted*. In the case study network agent 2 and agent 4 have self-loops with a link value of 1. The links are directed, therefore, every node has an in-degree (links pointing to a node) and an out-degree (links pointing from a node). The line weights change the degree centrality calculations from counting the links to summing up the link weighs. For this case study, we select out-degree centrality. Of course, the column “Treat as Symmetric” and “Treat as Simple” represents the symmetric degree centrality.

Differences ORA/UCINET

- Degree centrality calculation in UCINET offers the following options. 1) Treat data as symmetric. 2) Include Diagonal Values.
- The “treat as binay” is not available in UCINET as an option when calculating degree centrality.
- The results considering all characteristics are identical. Using the two UCINET option similar than the ORA primary measure settings also provides identical results.
- UCINET does not offer total degree centrality

Table A2: Case study out-degree centrality

	Consider All Characteristics	Treat as Symmetric	Treat as Binary	Ignore Self-Loops	Treat as Simple	UCINET
Agent1	0.200	0.200	0.333	0.240	0.400	20.0
Agent2	0.133	0.167	0.333	0.120	0.400	13.3
Agent3	0.033	0.367	0.167	0.040	0.600	3.3
Agent4	0.133	0.133	0.333	0.120	0.200	13.3
Agent5	0.033	0.033	0.167	0.040	0.200	3.3
Agent6	0.033	0.033	0.167	0.040	0.200	3.3
Network	0.127	0.304	0.100	0.168	0.400	12.7%

9.3 Betweenness Centrality

Betweenness centrality is affected by *symmetric/asymmetric* and *binary/weighted*. The importance of both characteristics is covered by the left component of the case study network. Asymmetric links limit the numbers of the possible shortest paths, e.g. agent 4 is reachable from agent 2, but not the vice versa. The link weights (which are interpreted in ORA as distances, see section 5) change the shortest path between agent 1 and agent 3. These two nodes are directly connected which is, of course, the shortest path in the binary case. In the weighted case the shortest path from agent 1 to agent 3 is the path via agent 2. Therefore, agent 2 gets a non-zero betweenness centrality score.

Differences ORA/UCINET

- UCINET automatically binarizes the network. Therefore, the “treat as binary” column of the ORA results is identical with the UCINET result.

Table A3: Case study betweenness centrality

	Consider All Characteristics	Treat as Symmetric	Treat as Binary	Ignore Self-Loops	Treat as Simple	UCINET
Agent1	0.000	0.000	0.000	0.000	0.000	0.0
Agent2	0.100	0.200	0.000	0.100	0.000	0.0
Agent3	0.100	0.200	0.100	0.100	0.200	10.0
Agent4	0.000	0.000	0.000	0.000	0.000	0.0
Agent5	0.000	0.000	0.000	0.000	0.000	0.0

Agent6	0.000	0.000	0.000	0.000	0.000	0.0
Network	0.080	0.160	0.100	0.080	0.200	10.0%

9.4 Closeness Centrality

The characteristics *symmetric/asymmetric*, *binary/weighted*, and *connected/disconnected* affect closeness centrality calculation. The first two characteristics are covered identically for betweenness centrality calculation in the case study network. Connected/disconnected is covered by two artifacts - by the two components and by unreachable nodes (e.g. agent 2 cannot reach agent 1). For the case study of closeness centrality we calculate out-closeness centrality.

Differences ORA/UCINET

- Both tools offer in-closeness and out-closeness centrality in case of directed networks.
- UCINET automatically binarizes the network. Therefore, the UCINET result fits the ORA result when the network is treated as binary.
- UCINET does not compute network level closeness centrality for unconnected graphs.
- The penalties for unreachable nodes are treated identically in UCINET and in ORA (see section 6).

Table A4: Case study out-closeness centrality

	Consider All Characteristics	Treat as Symmetric	Treat as Binary	Ignore Self-Loops	Treat as Simple	UCINET
Agent1	0.071	0.069	0.313	0.071	0.313	31.3
Agent2	0.052	0.071	0.238	0.052	0.313	23.8
Agent3	0.041	0.071	0.200	0.041	0.333	20.0
Agent4	0.041	0.066	0.200	0.041	0.294	20.0
Agent5	0.041	0.041	0.200	0.041	0.200	20.0
Agent6	0.041	0.041	0.200	0.041	0.200	20.0
Network	0.063	0.031	0.236	0.063	0.156	-

9.5 Eigenvector Centrality

The calculation of eigenvector centrality is influenced by the characteristics *allow/ignore self-loops*, *binary/weighted*, and *connected/disconnected*. In section 7 we discussed why eigenvector centrality automatically treats every network as symmetric. Allow/ignore self-loops and binary/weighted are covered by the algorithm itself. For disconnected networks we offer the calculation of eigenvector centrality per component.

Differences ORA/UCINET

- Eigenvector centrality is always calculated with symmetric networks in both tools. ORA as well as UCINET symmetrize the network for the calculations automatically.
- UCINET offers the option “Force majority of scores to be positive”. ORA does this automatically because we think that there is no useful case for deselecting this option.
- The node level results in ORA and UCINET are identical.
- UCINET tells you that the network level measure is “uninterpretable for disconnected graphs”. The score >100 % results in a different (and not optimal)

equation than we described in section 7 of this article. The calculations of ORA guarantees node and network level results in the range of [0-1].

- Eigenvector centrality per component avoids components with zero-values for all nodes.
- We discussed in section 7 that a network consisting of a single dyad results in the maximum possible eigenvector score. This makes agent 5 and 6 more important that agent 4. We selected this network to show the drawback of calculating eigenvector per component. If the other component(s) were larger the additional scaling would compensate for this artifact. Nevertheless, if your network consists of a couple of smaller components, we suggest removing all components with the size 1 or 2 before using the eigenvector centrality per component.

Table A5: Case study eigenvector centrality

	Consider All Characteristics	Treat as Symmetric	Treat as Binary	Ignore Self-Loops	Treat as Simple	Per Component	UCINET
Agent1	0.741	0.741	0.635	0.775	0.739	0.349	74.1
Agent2	0.580	0.580	0.880	0.531	0.739	0.273	58.0
Agent3	0.950	0.950	0.769	0.970	0.865	0.448	95.0
Agent4	0.460	0.460	0.482	0.419	0.399	0.217	46.0
Agent5	0.000	0.000	0.000	0.000	0.000	0.236	0.0
Agent6	0.000	0.000	0.000	0.000	0.000	0.236	0.0
Network	0.742	0.742	0.628	0.781	0.612	0.232	107.4%

9.6 Clustering Coefficient

We treat the clustering coefficient in a way that its results are affected by two characteristics, *allow/ignore self-loops* and *symmetric/asymmetric*. The obvious local clustering is covered in the left component (the triangle created by agents 1, 2, and 3). But also self-loops influence the results of the clustering coefficient (see section 8).

Differences ORA/UCINET

- When calculating the clustering coefficient UCINET takes other characteristics into consideration than ORA. Self-loops are always ignored, but on the other hand the characteristic binary/weighted changes the UCINET result. The characteristic symmetric/asymmetric is coverted by both tools. To generate the same outcome in ORA self-loops have to be ignored and in UCINET the network has to be binarized.
- UCINET does not scale the results for clustering coefficient into the range [0-1] or the range [0-100].
- UCINET offers the information that the transitivity measure in UCINET can be a weighted network level measure for the clustering coefficient.

Table A6: Case study clustering coefficient

	Consider All Characteristics	Treat as Symmetric	Treat as Binary	Ignore Self-Loops	Treat as Simple	UCINET
Agent1	0.500	0.667	0.500	0.500	1.000	1.500
Agent2	0.250	0.333	0.250	0.500	1.000	2.500
Agent3	0.333	0.500	0.333	0.167	0.333	0.167
Agent4	0.000	0.000	0.000	0.000	0.000	0.000
Agent5	0.000	0.000	0.000	0.000	0.000	0.000
Agent6	0.000	0.000	0.000	0.000	0.000	0.000
Network	0.181	0.250	0.181	0.194	0.389	1.389

10 Conclusions

In this article we discussed the handling of *weighted*, *asymmetric*, *self-looped*, and *disconnected* networks in ORA. We enumerated the impact of these characteristics on the five widely used measures in the field of social network analysis. The considerations in the context of the different measures resulted in different formulas for different combinations of the network characteristics. In the case studies we calculated the measures in ORA with different settings and compared the results with measure calculated by UCINET.

The causes for different results in ORA when applying different settings for the primary measure parameters but also the possible differences between results in ORA and results of other tools for social network analysis (e.g. UCINET) can be summarized to three underlying reasons. These three reasons can change the scores of measures on the node level and subsequently on the network level, but the first one (different scaling) does not influence the ranking of the nodes and is therefore a minor issue.

1. *Different scaling results in different scaled scores*. E.g., including the self-loops when calculating scaled degree centrality changes the scaling factor (see section 4.2).
2. *Different interpretation of a measure consequences altered algorithms*. E.g., self-loops are treated in ORA as one line which is counted for in-degree and for out-degree, but just one time for total degree. In contrast, it would be also possible to create total degree by summation of in- and out-degree. This would count self-loops twice which influence the scaling factor.

3. *Different handling of data artifacts changes the results of measure calculations*. The vast majority of formulas for non-simple networks which are enumerated in this article are not discussed in the original papers. Therefore, different groups of researchers can interpret the handling of network characteristics differently, e.g., unreachable nodes in closeness centrality. Often, there is no “right” and “wrong”, but the user has to know what is going on in different tools.

Table A7 shows the differences of handling weighted, asymmetric, self-looped, and disconnected networks in ORA and UCINET. When calculating measures there is one major difference based in the logic of the tools. UCINET offers handling of some network characteristics for different measures individually. In ORA the primary measure parameters are global settings which affect every single measure the same way. This should result in a more stable and consistent handling of the network characteristics. Looking at the results of calculations of the discussed measures using ORA and UCINET, we can summarize the following differences. About half of the different results for degree centrality can be reproduced identically in both tools. UCINET does not offer a “treat as binay” option or the calculation of the total degree centrality. When looking at betweenness and closeness centrality, the handling of weighted data in ORA is a very important and distinctive feature which changes the way shortest paths are calculated. The handling of unreachable nodes when calculating closeness centrality is treated identically as well as the scaling of eigenvector centrality on node level. The clustering coefficient is interpreted differently in ORA and in UCINET; ORA’s local density interpretation is un-weighted but includes self-loops, while UCINET considers weights but no self-loops.

Table A7: Differences ORA/UCINET

	ORA	**UCINET**
Handling of network characteristics	global parameters for all measures	individual handling in context of some measures
Scaling of network options	0 – 1 0 – 100	0 – 100
Measures scaled in 0-1 or 0-100 range	degree centrality betweenness centrality closeness centrality eigenvector centrality clustering coefficient	degree centrality betweenness centrality closeness centrality
Degree centrality options	allow/ignore self-loops symmetric/asymmetric binary/weighted	allow/ignore self-loops symmetric/asymmetric
Degree centrality variations	degree centrality in-degree centrality out-degree centrality total degree centrality	degree centrality in-degree centrality out-degree centrality
Betweenness centrality options	symmetric/asymmetric binary/weighted	symmetric/asymmetric
Closeness centrality options	symmetric/asymmetric binary/weighted	symmetric/asymmetric
Closeness centrality variations	closeness centrality in-closeness centrality out-closeness centrality	closeness centrality in-closeness centrality out-closeness centrality
Closeness centrality Penalty for unreachable nodes	$\|N\| \cdot w^*$	$\|N\| \cdot w^*$
Eigenvector centrality options	allow/ignore self-loops binary/weighted connected/disconnected	allow/ignore self-loops binary/weighted
Eigenvector centrality handling negative scores	avoid automatically	avoid optionally
Clustering coefficient options	allow/ignore self-loops symmetric/asymmetric	symmetric/asymmetric binary/weighted

Almost all differences between UCINET and ORA can be aligned with the primary measure settings in ORA or the parameters of some UCINET measures. When looking at these differences, one can state that ORA gives you more options to handle your non-simple networks more precisely the way you want to handle them. On the other hand, having more option also gives you more responsibility. Complex networks and different options of handling these networks require a deeper understanding of the applied network measures. This article should help you to better understand the different characteristics of network data and the implications of these characteristics to network measures, but it also should increase your awareness of your weighted, asymmetric, self-looped, and disconnected networks. Consequently, our final words with which to send you on your way to your network analytical projects are: Know your data, know your measures.

11 References

Anthonisse, J. M., 1971. The rush in a directed graph. Technical Report BN 9/71, Stichting Mathematisch Centrum, Amsterdam.

Bonacich, P., 1972. Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology 2, 113-120.

Borgatti, S.P., Everett, M.G., Freeman, L.C., 2002. Ucinet 6 for Windows. Cambridge, MA: Analytic Technologies.

Brandes, U., 2001. A faster algorithm for betweenness centrality. Journal of Mathematical Sociology 25 (2), 163–177.

Carley, K. M., 2002. “Smart Agents and Organizations of the Future.” In The Handbook of New Media, eds. Leah A. Lievrouw and Sonia Livingstone, pp. 206-220. Thousand Oaks, CA: Sage.

Carley, K. M., Reminga, J., Storrick, J., Columbus, D., 2010. ORA User’s Guide 2011. Carnegie Mellon University, School of Computer Science, Institute for Software Research, Technical Report, CMU-ISR-11-107

Freeman, L. C., 1977. A set of measures of centrality based on betweenness, Sociometry 40, 35-41.

Freeman, L. C., 1979. Centrality in social networks: Conceptual clarification. Social Networks 1, 215–239.

Granovetter, M. S., 1973. The Strength of Weak Ties, American Journal of Sociology 78 (6), 1360-1680.

Sabidussi, G., 1966. The centrality index of a graph. Psychometrika 31, 581–603.

Scott, J., 2000. Social Network Analysis, 2nd Edition, Sage Publications Ltd, Los Angeles.

Wasserman, S., Faust K., 1995. Social network analysis. Methods and applications. Cambridge University Press.

Watts, D., Strogatz S., 1998. Collective dynamics of small world networks. Nature, Vol. 393, 440–442.